1. Redundancy Management

1.1 Redundancy Management CSC Introduction

1.1.1 Redundancy Management CSC Overview

The Redundancy Management CSC monitors, and to a lesser extent, maintains the health of the RTPS. It does this by monitoring the health of both the software and hardware in the system. *If failures are detected, and a recovery mechanism is in place, Redundancy Management implements the recovery*. All failures cause generation of a System Message.

Redundancy Management also manages the System Configuration Table(SCT) during operations. This table specifies the hardware and software configuration, both logical and physical. It allocates resources to Test Sets, which then support specific Activities. The static portion, and initial values for much of the dynamic part of the SCT are generated off-line, then loaded by Redundancy Management at initialization. Redundancy Management then maintains the dynamic portion of the table and makes all data available to displays and other applications.

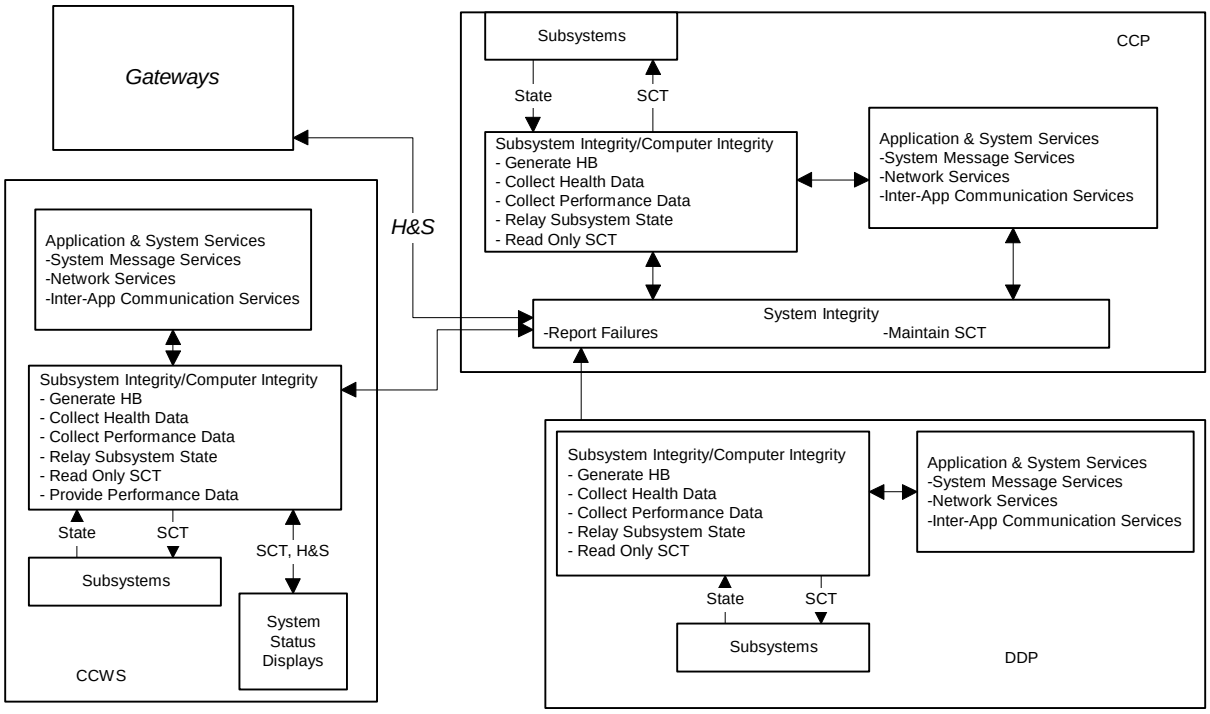

Figure 1. Redundancy Management Conceptual Data Flow Diagram

1.1.2 Redundancy Management CSC Operational Description

As shown in Figure 2, Redundancy Management is composed of five parts: *Set Integrity*, System Integrity, Subsystem Integrity, Computer Integrity and System Configuration Table. Each Integrity monitors the integrity of its parts and reports the results of this analysis to a higher level. The collected integrity is reflected in the SCT.

Redundancy Management executes in the DDPs, CCPs, CCWSs, and the CM Server. *Subsystem and Computer Integrity equivalents execute in the Gateways*, but are not part of this CSC. *Set Integrity executes in the Set Master CCWS*. The SCT is available to any computer that executes Redundancy Management.

Computer Integrity executes in each Computer, monitors the health and status of the computer, and records standard hardware performance data. It makes the health data available to any local Subsystem Integrity operation. Both the health and performance data are provided to System Integrity. Computer Integrity also generates the heartbeat that is delivered to System Integrity and Set Integrity.

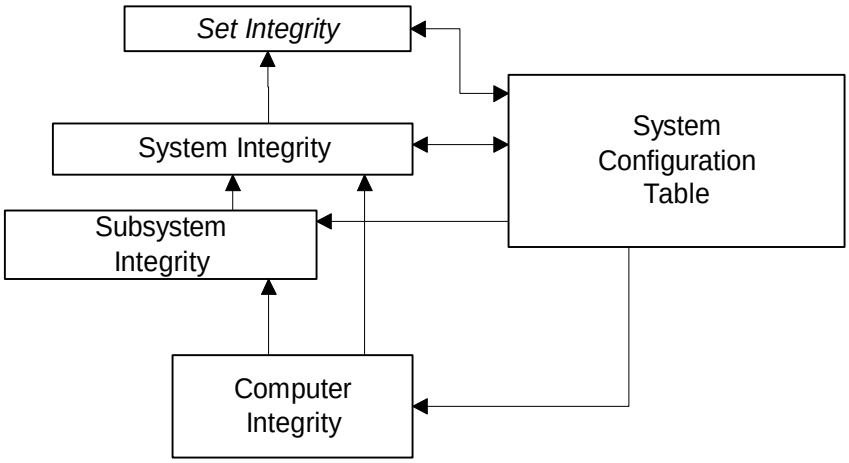

Figure 2. Logical Redundancy Management Organization

Subsystem Integrity executes in each Subsystem and uses the Computer health and information provided from applications in the Subsystem to generate Subsystem Health. This health and status report is provided to System Integrity. Using a combination of the Health and Status report and information from other subsystems, System Integrity makes a determination on the health of the Subsystem. Any changes in the health and status are recorded in the System Configuration Table, which is viewable on the System Status Display.

Based on collected data, System Integrity determines the health of each subsystem and keeps the SCT current. It also provides general health of the Test Set to Set Integrity. System Integrity executes in the CCP.

1.2 Redundancy Management CSC Specifications

1.2.1 Redundancy Management CSC Groundrules

- Definitions
    - An Activity is a named operation performed as part of a test. It may consist of lower level Activities.
    - Computer is defined to be a physical box that may contain more than one CPU or Processor
        - Computers communicate with other Computers over the Network
        - Any peripherals in the same physical box such as a disk drive, memory, cards are also considered to be part of the Computer.
    - CPU and Processor are used interchangeably to refer to actual compute engine of a computer. Many modern Computers have more than one CPU.
- *While not a Thor requirement, there will be a need in the IDE to execute multiple Subsystems on a single Computer.*

- Redundant subsystems are actually composed of two Subsystems, one designated as active, one designated as standby. In this document, Subsystem refers to the Active or Standby Subsystem, not the combined pair.
- Cyclic rates are independent of Activity in which the subsystem is involved.
- If COTS is used, most or all SI to SSI communication will be SNMP, not FDs.
- Executable code with an API is provided for the CCPs, DDPs, CCWSs, and CM. This replaces the Application Services provided API identified in the System Integrity thread.
- An application independent heartbeat is sufficient. Application health will be reported through Health and Status data rather than receipt of a heartbeat.
- Dependencies:
  - Ideally, System Integrity will use Data Fusion and Constraint Management services to build failure data from raw measurements. This would allow user configuration and tailoring of the failure criteria. More analysis of whether this is possible will be done during early design phases.
  - Interface from network services to update the logical to physical processor map.
  - A Thor user of the Subsystem API.
  - Interface for generation of System Messages (System Services)
- External Interfaces:
  - *Gateways will supply FDs as required by System Integrity*.
  - *TBD FDs*
  - System Messages
    - Missed Heartbeat
    - Computer State Change
    - Subsystem State Change

1.2.2 Redundancy Management CSC Functional Requirements

The Functional Requirements for System Integrity are arranged in the following major/minor functions:

1. Set Integrity

2. System Integrity
3. Subsystem Integrity
4. Computer Integrity
5. System Configuration Table

1 Set Integrity

Set Integrity provides the equivalent of System Integrity, but for all Test Sets in the system and any computers not currently assigned to a Test Set

2 System Integrity

System Integrity evaluates the operations on computers and subsystems. It provides data for display at the System Status Display, and reports any errors or failures as System Messages. *When automatic recovery such as switchover is possible, it determines when to recover and directs the recovery.*

2.1 System Integrity will be a Redundant Subsystem

2.2 There will be one System Integrity *Redundant Pair* in each Test Set.

2.3 System Integrity will execute in the CCP.

2.4 If the received Computer Heartbeat contains the next expected Heartbeat and current State of the Computer is Go, no change will be made to the SCT. Note that the next expected heartbeat will typically be 1 greater than the current heartbeat, but it may be a rollover.

2.5 If the received Computer Heartbeat contains a Heartbeat other than the next expected heartbeat, a Missed Heartbeat System Message will be generated.

2.6 If two consecutive Computer Heartbeats are missed, the Computer will be placed in a No Go State. Note that the missing heartbeats must be detected without receiving a packet from the computer.

2.7 If a Computer is placed in a No Go State, all Subsystems on the Computer will be placed in a No Go State.

2.8 If a Computer is placed in a No Go State, a Computer State Change System Message will be generated.

2.9 Upon receipt of a Subsystem Health Message, System Integrity will update the SCT to reflect any changes in the state of the subsystem.

2.10 If a Subsystem is placed in a No Go State, a Subsystem State Change System Message will be generated.

2.11 System Integrity will use stuff other than the Subsystem Health Message and Computer Heartbeat to determine the Subsystem Health.

2.12 If an Active Redundant Subsystem is placed in a No Go State, System Integrity will trigger the Standby Subsystem to transit to the Active Role.

3 Subsystem Integrity

Subsystem Integrity monitors health at the subsystem level and relays the subsystem health to System Integrity.

3.1 Subsystem Integrity will be a part of each non-Gateway Subsystem.

3.2 Subsystem Integrity will provide an API that allows a Process to record changes to Health Data as detected by the Process.

3.3 Subsystem Integrity will provide an API that allows a Subsystem to record changes to Health Data as detected by the Subsystem.

3.4 Subsystem Integrity will notify System Integrity of any change to Health Data for the Subsystem as required for the Subsystem in its current State and Role.

3.5 If a Process designated by the SCT as Critical Fails, Subsystem Integrity will request that System Integrity initiate a Switchover.

3.6 For all subsystems, the following health and status data will be supplied: (provided by the subsystem through the API)

- 3.6.1 State:
- 3.6.2 Switchover Status: (Active, Standby)
- 3.6.3 Commands Received Since Go
- 3.6.4 Invalid Commands Received Since Go
- 3.6.5 Process Health

3.7 For the CCP, the following health data will be supplied:

- 3.7.1 Process State (Running, Not Running) for (Critical/Non-Critical), (System/Application) processes

3.8 For the DDP, the following health data will be supplied:

3.9 For the CCWS, the following health data will be supplied: no unique health data in Thor.

4 Computer Integrity

Computer Integrity monitors each computer and relays the health of the computer to System Integrity.

4.1 Only one copy of Computer Integrity will execute on a Computer regardless of the number of Subsystems executing.

4.2 Computer Integrity will execute on all powered on processors, regardless of whether any subsystems are loaded.

4.3 Upon successful start, Computer Integrity will broadcast its serial number and network address.

4.4 Computer Integrity will issue a Heartbeat Message at a periodic rate as specified by the Subsystems on the Computer and the State and Role of those Subsystems. The Computer Integrity Performance Requirements specify the rates to be used for Thor.

4.5 If more than one Subsystem is on a Computer, the Heartbeat Message will be generated at the highest rate required by the Subsystems and their respective States

4.6 Computer Integrity will provide the following performance data periodically at the rate required by the Subsystem on the Computer, and their States and Roles. The Computer Integrity Performance Requirements specify the rates to be used for Thor:

- 4.6.1 Average percent CPU Utilized over the period
- 4.6.2 Average percent memory available over the period
- 4.6.3 Network throughput during the period per unit time for each network to which the Computer is attached.
- 4.6.4 Network interrupts received during the period per unit time for each network to which the computer is attached.
- 4.6.5 Number of Network Errors since Go for each network to which the computer is attached.
- 4.6.6 Average disk utilization during the period.
- 4.6.7 Number of disk accesses since Go
- 4.6.8 Number of Disk Errors since Go

5 System Configuration Table

The System Configuration Table provides a logical and physical map of the Set. Within a Test set, the data is constrained to the computers and subsystems in the Test Set. A system-wide SCT is maintained for Set Master operations. Figure 3 provides an illustration of SCT data and relationships within and external to the SCT.

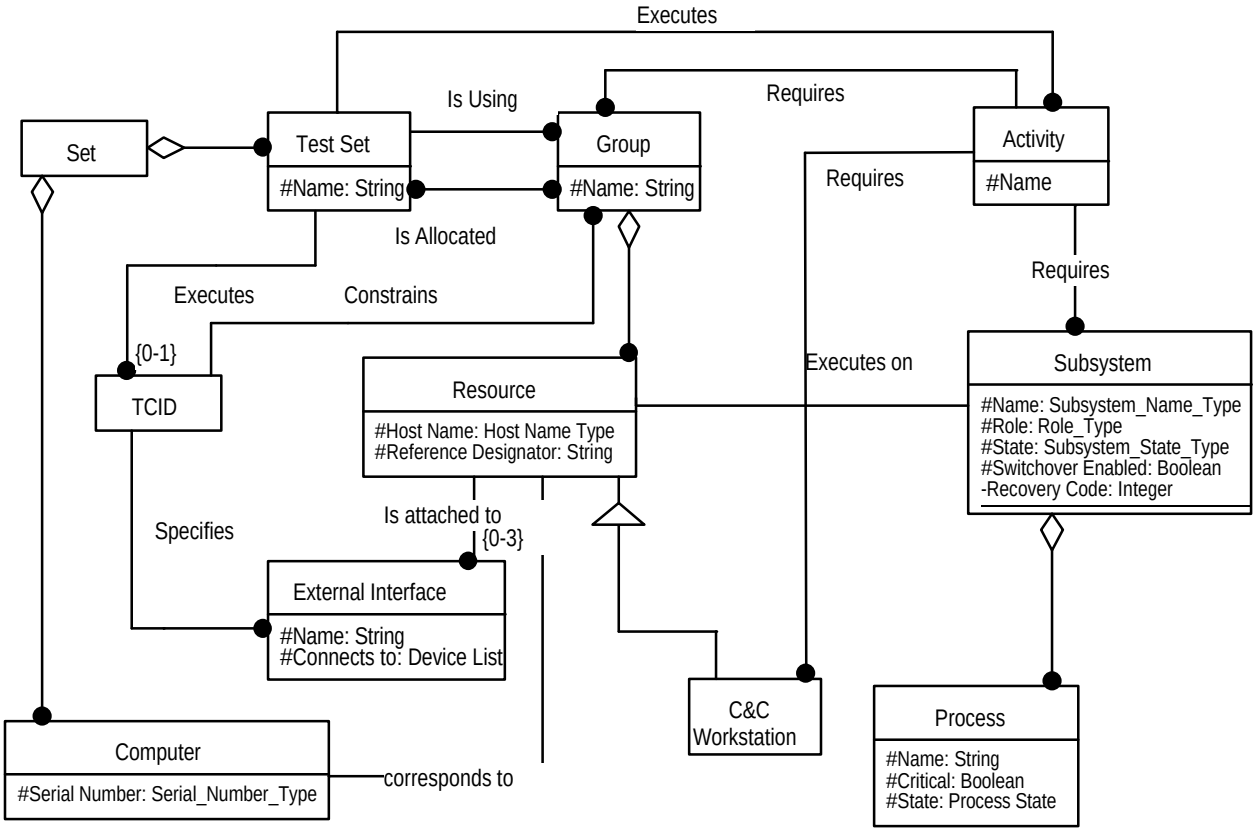

Figure 3 - SCT Class Structure

Bold items are defined in a different 5.x requirement

5.1 The SI CSC will update the network configuration to reflect changes in the SCT.

5.2 The SI CSC will provide an API to allow other applications to retrieve SCT data

5.3 The System Configuration Table will make the following data available for each **Set**:

- 5.3.1 The Name of the Set
- 5.3.2 The **Test Sets** that make up the Set

5.3.3 The logical **Resources** that make up the Set.

5.4 The SCT will make the following data available for each logical **Resource**:
- 5.4.1 Host Name (i.e., LCCGWGSE1)
- 5.4.2 Reference Designator (i.e., 130A1)
- 5.4.3 Attached **External Interfaces**
- 5.4.4 Physical **Computer**

5.5 The SCT will make the following data available for each **Test Set**
- 5.5.1 Name
- 5.5.2 **Group**s allocated to the Test Set
- 5.5.3 **Group**s in use by the Test Set
- 5.5.4 CCWS **Resources** allocated to the test set
- 5.5.5 CCWS **Resources** in use by the Test Set
- 5.5.6 TCID being executed in the Test Set
- 5.5.7 **Activities** being executed in the Test Set

5.6 The SCT will make the following data available for each **External Interface**
- 5.6.1 Name (i.e., GSE, LDB)

5.7 The SCT will make the following data available for each physical **Computer**
- 5.7.1 Serial Number

5.8 The SCT will make the following data available for each **Group**:
- 5.8.1 Name
- 5.8.2 Type (e.g., Control Group, Front End Zone)
- 5.8.3 The **Resources** assigned to the Group

5.9 The SCT will make the following data available for each **Activity**:
- 5.9.1 Name (i.e., S0007)

5.10 The SCT will make the following data available for each **Subsystem**:
- 5.10.1 Name (i.e., GS1A, GS1S)
- 5.10.2 Role (i.e., Active, Standby, Hot Spare)
- 5.10.3 State (e.g., In Configuration, Platform Initialized, SCID Initialized, Loaded, Communicating, Go, In ORT)
- 5.10.4 **Resource** on which the Subsystem is executing

1.2.3 Redundancy Management CSC Performance Requirements

Provide a bulleted list of performance requirements that describe the performance requirements for the CSC. These can be described in terms of the threads that are involved and their required performance. The performance requirements must be stated in quantifiable, measurable parameters. These requirements should be stated in the same order as major/minor functions above and should use the same convention for major/minor function headings.

1 System Integrity

1.1 System Integrity will detect a missing Computer Heartbeat message within 1 cycle of the expected arrival time.

This means that if the HB is scheduled for frequency of 10ms, that SI will detect the missing packet within 10ms of its expected arrival time. With a failure declared at two missed cycles, this allows for a maximum 30ms detected failure time: If the failure occurs immediately following a heartbeat, 10ms elapses before the first HB is not generated, 10 more before the second is not generated, 10 more before SI realizes that two have been missed. In order to not declare false alerts, and allow for network delays, SI will probably be scheduled to look for the HB somewhere late in the cycle.

2 Subsystem Integrity

– None.

3 Computer Integrity

3.1 For Thor, the following rates will be used for health and status reporting:

		Active Subsystem Xmit Rate		Standby Subsystem Xmit Rate	
ID	Subsystem	HC FD	Status FDs *	HC FD	Status FDs *
1.1.1	GSE G/W	SSR	1/Sec	1/Sec	1/Sec
1.1.2	PCM DL G/W	SSR	1/Sec	1/Sec	1/Sec
1.1.3	SSME GW	SSR	1/Sec	1/Sec	1/Sec
1.1.4	LDB G/W	SSR	1/Sec	1/Sec	1/Sec
1.1.5	PCM UPLK G/W	SSR	1/Sec	1/Sec	1/Sec
1.1.6	Consolidated G/W	SSR	1/Sec	1/Sec	1/Sec
1.1.7	DDP	SSR	1/Sec	1/Sec	1/Sec
1.1.8	CCPs	SSR	1/Sec	1/Sec	1/Sec
1.1.9	Ops CM Server	DSR	1/Sec	1/Sec	1/Sec
1.1.10	SDC	TBD	1/Sec	1/Sec	1/Sec
1.1.11	C & C W/Ss	DSR	1/Sec	1/Sec	1/Sec
1.1.12	Unassigned Computer	1/Sec	N/A	N/A	N/A

1.1 For Thor, the SSR will be 100 Hz. (10ms cycle)
1.2 For Thor, the DSR will be 10 Hz. (100 ms cycle)

2 System Configuration Table

2.1 Correct System Configuration data will be provided to an API user within TBD ms of the API invocation.

The intent of this requirement is to allow distributed maintenance of the SCT if performance constraints permit.

2.2 A change in the SCT will be visible on all processors within 10 ms of the change.

1.2.4 Redundancy Management CSC Interfaces Data Flow Diagrams

This section describes the interfaces to the RM CSC.

- Subsystem CSCIs provide application health data and can read the SCT as required. The provided health data is used in subsequent SCT updates.
- Ideally, RM will use both Data Fusion and Contraint Management to detect failures in the system. Exact use of these capabilities will be examined in the design phase.
- System Integrity uses System Services to deliver system messages that notify operators of subsystem failures.
- Status is visible to operators through System Viewers that can extract data from the SCT.
- The initial values for the SCT are provided through file(s) exported from a TBD COTS Office tool such as Microsoft Access. Actual tool used for Thor will be specified in the design phase.

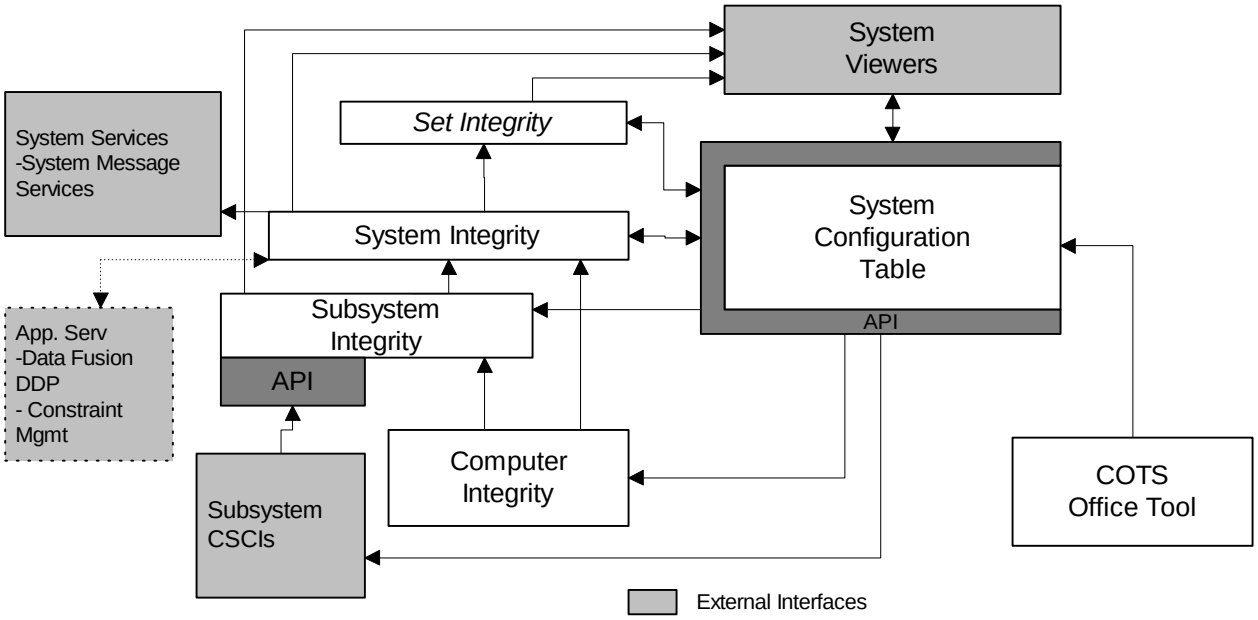

Figure 4 - RM CSC Data Flow

Table 1. External Interface Table

	API			
CSCI	**Subsystem Integrity**	**SCT Read**	**SCT Write**	**Provides I/F**
System Services				X
Application Services	*X*	X		X
Data Distribution and Processing	*X*	X		X
System Viewers		X	X	
Command Support	X	X		
Orbiter Computation Facility				
System Control				
Ops Configuration Manager CSC	X	X		
System Diagnostics				
Common Gateway Services				
Consolidated System Gateway Services				
GSE Gateway Services				
LDB Gateway Services				
PCM D/L Gateway Services				
Uplink Gateway Services				
Sim Gateway Services				
CLCS Development Environment				
System Build				
DBSAFE				
Test Build and Control				
Data Recording/Archival and Retrieval				
SDS Services				
Common App Support				

	API			
CSCI	**Subsystem Integrity**	**SCT Read**	**SCT Write**	**Provides I/F**
SLWT	*X*			
HMF				
Integrated Operations				
APU				
HYD				
COMM				
NAV				
DPS				
DPSME				
ECLSS				
ECS				
EPDC				
GLS				
GUID				
HAZ GAS				
BHYD				
BAPU				
INST				
Swing Arm				
LH2				
LO2				
MSTR				
MEQ				
MPS				
SSME				
OMS/RCS				
FLT Controls				
PLDTEST				
PRSD/FC				
SRSS				
Water				
CITE				
CCS				
Near Real-Time Advisory				
Data Support Tools				
Italics = Not in Thor X = Uses API, or Provides Service				